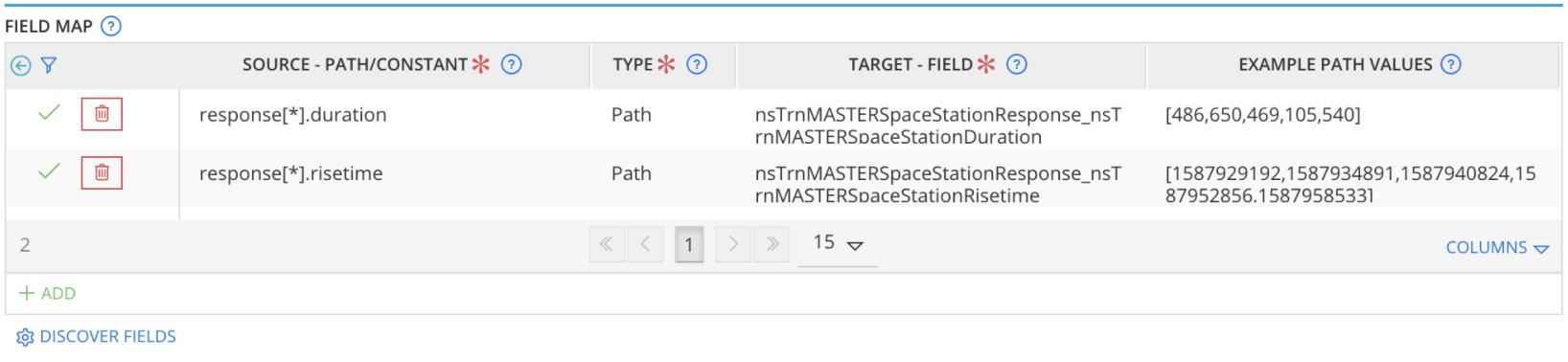
**Business Requirement Summary:**

Integrating with a 3rd-party system is a common requirement for ERP systems. This exercise is unrelated to the capstone. As a standalone exercise, it’s more light-hearted in nature. Using this API <http://open-notify.org/Open-Notify-API/ISS-Pass-Times/>, we will build a callout that passes in a latitude and longitude and receives data on when the International Space Station will pass overhead and how long it will be visible.

|  |  |
| --- | --- |
| **Task** | **Notes** |
| Create Response Table | Name: SpaceStationResponse  Add Fields (they already exist):   * nsTrnMASTERSpaceStationDuration * nsTrnMASTERSpaceStationRisetime |
| Create Subtable Data Item | Name: SpaceStationResponse (same as table above)  Type: Subtable |
| Create Request Table | Add Fields:   * nsTrnMASTERLatitude (already exists) * nsTrnMASTERLongitude (already exists) * SpaceStationResponse (created in the previous step) |
| Create Endpoint Configuration | 1. Define Header 2. Define Request    1. Add Query Params       1. lat / Field / nsTrnMASTERLatitude       2. lon / Field / nsTrnMASTERLongitude 3. Define Response    1. Define Response Set       1. Click on “Configure Response Set”       2. Add a row to the “Configured Response Codes” subtable          1. Response Code = 2xx       3. Click on “Manage Response Codes”       4. Edit the “2xx” line       5. Paste this in the Response Body:   {  "message": "success",  "request": {  "passes": 5,  "altitude": 100,  "datetime": 1587923027,  "latitude": 30.11,  "longitude": 101.22  },  "response": [  {  "duration": 486,  "risetime": 1587929192  },  {  "duration": 650,  "risetime": 1587934891  },  {  "duration": 469,  "risetime": 1587940824  },  {  "duration": 105,  "risetime": 1587952856  },  {  "duration": 540,  "risetime": 1587958533  }  ]  }   * + 1. Click on “Discover Fields”     2. Edit the Field Map to tell the mapping where to put the returned values. In this case, we want to place the returned values into the SpaceStationResponse subtable. Click on the “?” next to Source - Path/Constant to better understand how the mappings work (see Image 1 below)  1. In the Endpoint Configuration, in the Response page, attach the Response Set Name |
| Create Application | This is the application we will use to define the parameters and call the logic block.  Layout:   * Row A - Latitude, Longitude * Row B - SpaceStationResponse subtable   Subtable Fields   * Hide On Table = False for Duration and Risetime |
| Create Logic Block | This is the logic block that calls the endpoint and assign the returned values subtable in the UI.  Table: SpaceStationRequest  Type: Transaction  Logic:   * Check if Latitude and Longitude are not empty * If so, call the external endpoint. The *Table - To* can be whatever you would like (e.g. *SpaceStationCallResponse*). This is how you will reference the response data. * In our endpoint config, you told it to map the response to the SpaceStationResponse subtable.   + Loop through that subtable   + Insert into the *SpaceStationResponse.* :info:This may be confusing. We are mapping from SpaceStationResponse subtable to the SpaceStationResponse subtable. However, the first reference is to the returned data from the endpoint call. The second reference (being inserted into) is to the subtable in the record on the UI. |
| Add Row Action | Add a row action in your application:   * Logic Block = LB you just created * Display Name = Get Data * Link Type = Logic Block in Trans Boundary |
| :warning:Checkpoint | If the Latitude and Longitude are filled in, rows are added to the SpaceStationRequest table (see Imagine 2 below) |

**Image 1**



**Image 2**

